Laporan Hasil Praktikum

Algortima Dan Sistem Data

Praktikum 5



Angel Chelssa Leoniy Eka Permatasari

244107020202

1E

Program Studi Teknologi Informasi

Jurusan Teknik Informatika

POLINEMA

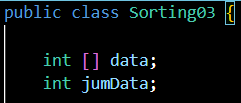
2025

**Praktikum 1**

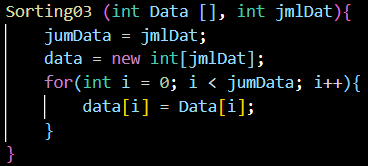
1. **SORTING – BUBBLE SORT**
2. Buat folder baru dengan nama Praktikum05.



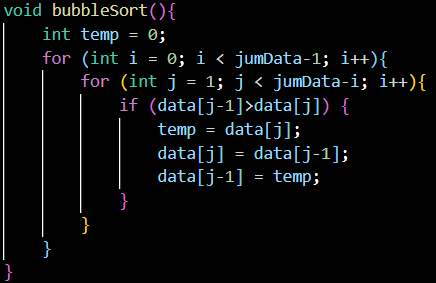
1. Buat class Sorting, kemudian tambahkan atribut sebagai berikut:



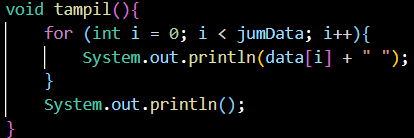
1. Buatlah konstruktor dengan parameter Data[] dan jmlDat



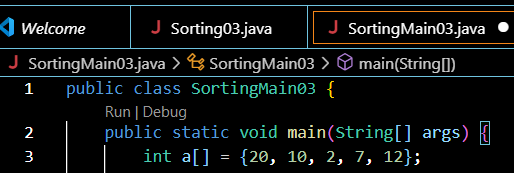
1. Buatlah method bubbleSort bertipe void dan deklarasikan isinya menggunakan algoritma Bubble Sort



1. Buatlah method tampil bertipe void dan deklarasikan isi method tersebut



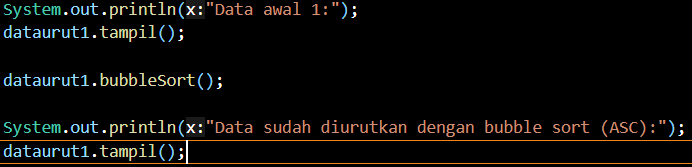
1. Buat class SortingMain kemudian deklarasikan array dengan nama a[] kemudian isi array tersebut



1. Buatlah objek baru dengan nama dataurut1 yang merupakan instansiasi dari class Sorting, kemudian isi parameternya



1. Lakukan pemanggilan method bubbleSort dan tampil



**Kode program**

**Kode sorting03.java**

|  |  |
| --- | --- |
| public class Sorting03 {      int [] data;      int jumData;      Sorting03 (int Data [], int jmlDat){          jumData = jmlDat;          data = new int[jmlDat];          for(int i = 0; i < jumData; i++){              data[i] = Data[i];          }      }      void bubbleSort(){          int temp = 0;          for (int i = 0; i < jumData-1; i++){              for (int j = 1; j < jumData-i; i++){                  if (data[j-1]>data[j]) {                      temp = data[j];                      data[j] = data[j-1];                      data[j-1] = temp;                  }              }          }      }      void tampil(){          for (int i = 0; i < jumData; i++){              System.out.println(data[i] + " ");          }          System.out.println();      }  } |  |

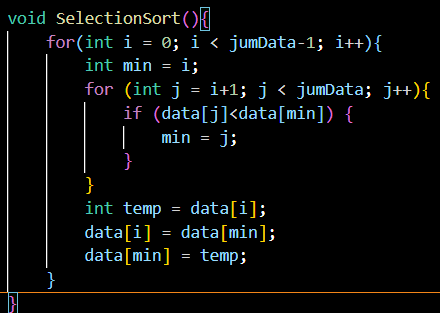
**Kode sortingMain03.java**

|  |
| --- |
| public class SortingMain03 {      public static void main(String[] args) {          int a[] = {20, 10, 2, 7, 12};            Sorting03 dataurut1 = new Sorting03(a, a.length);          System.out.println("Data awal 1:");          dataurut1.tampil();          dataurut1.bubbleSort();          System.out.println("Data sudah diurutkan dengan bubble sort (ASC):");          dataurut1.tampil();      }  } |

**Hasil kode**



1. **SORTING – SELECTION SORT**
2. Pada class Sorting yang sudah dibuat di praktikum sebelumnya tambahkan method SelectionSort yang mengimplementasikan pengurutan menggunakan algoritma selection sort.



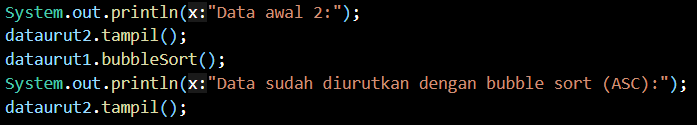
1. Deklarasikan array dengan nama b[] pada kelas SortingMain kemudian isi array tersebut



1. Buatlah objek baru dengan nama dataurut2 yang merupakan instansiasi dari class Sorting, kemudian isi parameternya



1. Lakukan pemanggilan method SelectionSort dan tampil



**Kode program**

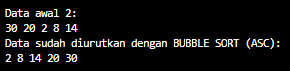
**Kode Sorting03.java**

|  |
| --- |
| public class Sorting03 {      int [] data;      int jumData;      Sorting03 (int Data [], int jmlDat){          jumData = jmlDat;          data = new int[jmlDat];          for(int i = 0; i < jumData; i++){              data[i] = Data[i];          }      }      void bubbleSort(){          int temp = 0;          for (int i = 0; i < jumData-1; i++){              for (int j = 1; j < jumData-i; i++){                  if (data[j-1]>data[j]) {                      temp = data[j];                      data[j] = data[j-1];                      data[j-1] = temp;                  }              }          }      }      void SelectionSort(){          for(int i = 0; i < jumData-1; i++){              int min = i;              for (int j = i+1; j < jumData; j++){                  if (data[j]<data[min]) {                      min = j;                  }              }              int temp = data[i];              data[i] = data[min];              data[min] = temp;          }      }      void tampil(){          for (int i = 0; i < jumData; i++){              System.out.println(data[i] + " ");          }          System.out.println();      }  } |

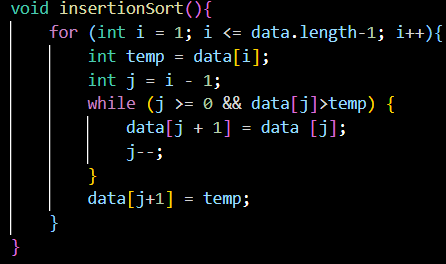
**Kode SortingMain03.java**

|  |
| --- |
| public class SortingMain03 {      public static void main(String[] args) {          int a[] = {20, 10, 2, 7, 12};          int b[] = {30, 20, 2, 8, 14};          Sorting03 dataurut1 = new Sorting03(a, a.length);          Sorting03 dataurut2 = new Sorting03(b, b.length);            System.out.println("Data awal 1:");          dataurut1.tampil();          dataurut1.bubbleSort();          System.out.println("Data sudah diurutkan dengan bubble sort (ASC):");          dataurut1.tampil();          System.out.println("Data awal 2:");          dataurut2.tampil();          dataurut1.bubbleSort();          System.out.println("Data sudah diurutkan dengan bubble sort (ASC):");          dataurut2.tampil();      }  } |

**Hasil kode**



1. **SORTING – INSERTION SORT**
2. Pada class Sorting yang sudah dibuat di praktikum sebelumnya tambahkan method insertionSort yang mengimplementasikan pengurutan menggunakan algoritma insertion sort.



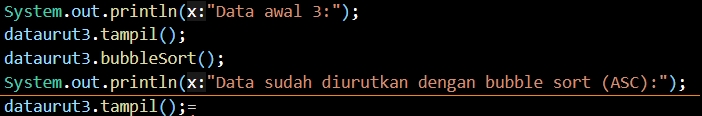
1. Deklarasikan array dengan nama c[] pada kelas SortingMain kemudian isi array tersebut



1. Buatlah objek baru dengan nama dataurut3 yang merupakan instansiasi dari class Sorting, kemudian isi parameternya



1. Lakukan pemanggilan method insertionSort dan tampil



**Kode program**

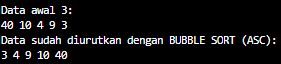
**Kode Sorting03.java**

|  |
| --- |
| public class Sorting03 {      int[] data;      int jumData;      public Sorting03(int Data[], int jmlDat) {          jumData = jmlDat;          data = new int[jmlDat];          for (int i = 0; i < jumData; i++) {              data[i] = Data[i];          }      }      void bubbleSort() {          int temp;          for (int i = 0; i < jumData - 1; i++) {              for (int j = 1; j < jumData - i; j++) {                  if (data[j - 1] > data[j]) {                      temp = data[j];                      data[j] = data[j - 1];                      data[j - 1] = temp;                  }              }          }      }      void selectionSort() {          for (int i = 0; i < jumData - 1; i++) {              int min = i;              for (int j = i + 1; j < jumData; j++) {                  if (data[j] < data[min]) {                      min = j;                  }              }              int temp = data[i];              data[i] = data[min];              data[min] = temp;          }      }      void insertionSort() {          for (int i = 1; i < jumData; i++) {              int temp = data[i];              int j = i - 1;              while (j >= 0 && data[j] > temp) {                  data[j + 1] = data[j];                  j--;              }              data[j + 1] = temp;          }      }      void tampil() {          for (int i = 0; i < jumData; i++) {              System.out.print(data[i] + " ");          }          System.out.println();      }  } |

**Kode SortingMain03.java**

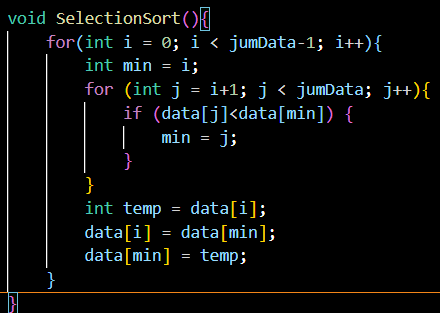
|  |
| --- |
| public class SortingMain03 {      public static void main(String[] args) {          int a[] = {20, 10, 2, 7, 12};          int b[] = {30, 20, 2, 8, 14};          int c[] = {40, 10, 4, 9, 3};          Sorting03 dataurut1 = new Sorting03(a, a.length);          Sorting03 dataurut2 = new Sorting03(b, b.length);          Sorting03 dataurut3 = new Sorting03(c, c.length);          System.out.println("Data awal 1:");          dataurut1.tampil();          dataurut1.bubbleSort();          System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC):");          dataurut1.tampil();          System.out.println("\nData awal 2:");          dataurut2.tampil();          dataurut2.bubbleSort();          System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC):");          dataurut2.tampil();          System.out.println("\nData awal 3:");          dataurut3.tampil();          dataurut3.bubbleSort();          System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC):");          dataurut3.tampil();      }  } |

**Hasil kode**

****

**Pertanyaan**

1. untuk menukar posisi dua elemen array jika elemen sebelumnya lebih besar dari elemen sesudahnya. bagian tersebut adaalah proses Bubble Sort (pengurutan secara ascending/menaik).
2. Kode yang merupakan algoritma pencarian nilai minimum pada selection sort adalah



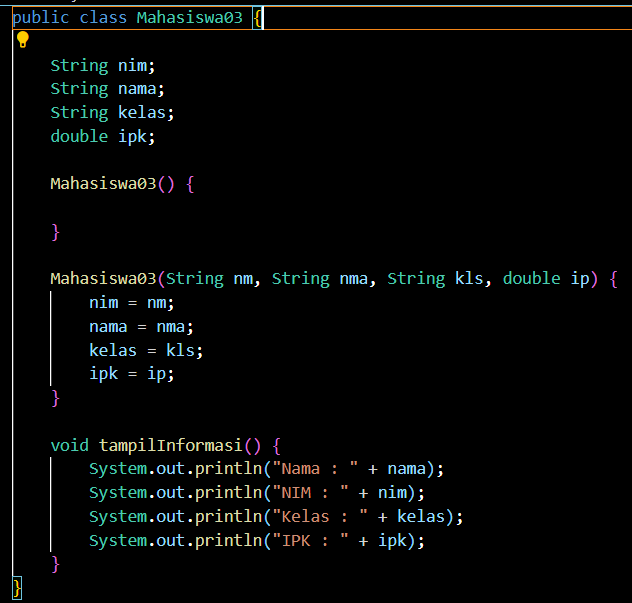
1. Kondisi while tersebut digunakan untuk menentukan berapa banyak elemen yang harus digeser ke kanan, agar nilai temp bisa disisipkan di posisi yang benar dalam proses pengurutan secara insertion sort.
2. Untuk menggeser elemen array ke kanan satu posisi agar memberikan ruang kosong bagi elemen yang akan disisipkan.

**Paraktikum 2**

1. **Mengurutkan data mahasiswa**
2. **Langkah” praktikum**
3. Buatlah class dengan nama Mahasiswa03.java

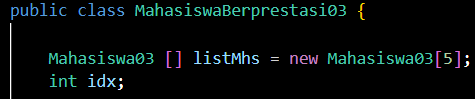


1. Untuk lebih jelasnya class tersebut dapat dilihat pada potongan kode di bawah ini

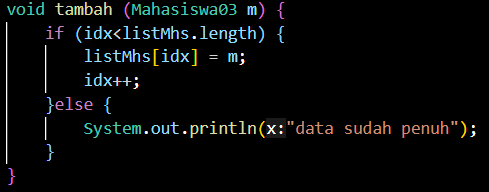


1. Buat class MahasiswaBerprestasi03.java seperti di bawah ini!

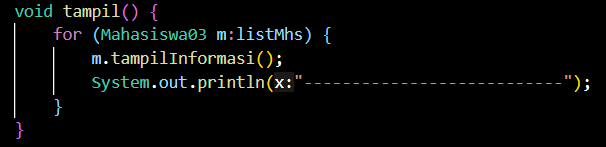




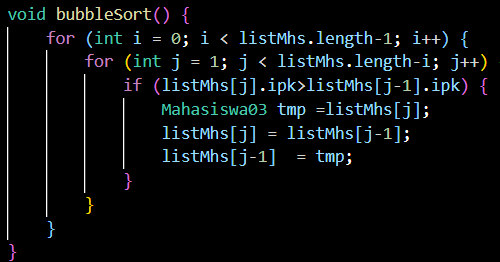
1. Tambahkan method tambah() di dalam class tersebut! Method tambah() digunakan untuk menambahkan objek dari class Mahasiswa ke dalam atribut listMhs



1. Tambahkan method tampil() di dalam class tersebut! Method tampil() digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada di dalam class tersebut! Perhatikan penggunaan sintaks for yang agak berbeda dengan for yang telah dipelajari sebelumnya, meskipun secara konsep sebenarnya mirip.

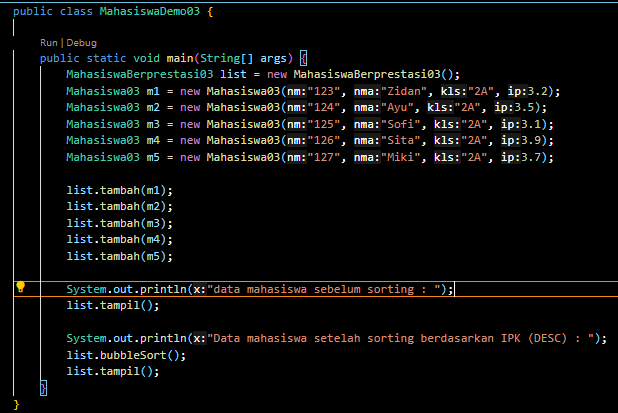


1. Tambahkan method bubbleSort() di dalam class tersebut!



1. Buat class MahasiswaDemo03.java, kemudian buatlah sebuah objek MahasiswaBerprestasi dan buatlah 5 objek mahasiswa kemudian tambahkan semua objek mahasiswa tersebut dengan memanggil fungsi tambah pada objek MahasiswaBerprestasi. Silakan dipanggil fungsi tampil() untuk melihat semua data yang telah dimasukan, urutkan data tersebut dengan memanggil fungsi bubbleSort() dan yang terakhir panggil fungsi tampil kembali.





**Kode program**

**Kode Mahasiswa03.java**

|  |
| --- |
| public class Mahasiswa03 {      String nim;      String nama;      String kelas;      double ipk;      Mahasiswa03() {      }      Mahasiswa03(String nm, String nma, String kls, double ip) {          nim = nm;          nama = nma;          kelas = kls;          ipk = ip;      }      void tampilInformasi() {          System.out.println("Nama : " + nama);          System.out.println("NIM : " + nim);          System.out.println("Kelas : " + kelas);          System.out.println("IPK : " + ipk);      }  } |

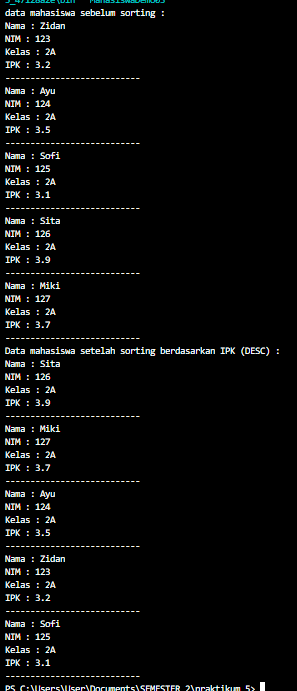
**Kode MahasiswaBerprestasi03.java**

|  |
| --- |
| public class MahasiswaBerprestasi03 {      Mahasiswa03 [] listMhs = new Mahasiswa03[5];      int idx;      void tambah (Mahasiswa03 m) {          if (idx<listMhs.length) {              listMhs[idx] = m;              idx++;          }else {              System.out.println("data sudah penuh");          }      }      void tampil() {          for (Mahasiswa03 m:listMhs) {              m.tampilInformasi();              System.out.println("---------------------------");          }      }      void bubbleSort() {          for (int i = 0; i < listMhs.length-1; i++) {              for (int j = 1; j < listMhs.length-i; j++) {                  if (listMhs[j].ipk>listMhs[j-1].ipk) {                      Mahasiswa03 tmp =listMhs[j];                      listMhs[j] = listMhs[j-1];                      listMhs[j-1]  = tmp;                  }              }          }      }  } |

**Kode MahasiswaDemo03.java**

|  |
| --- |
| public class MahasiswaDemo03 {      public static void main(String[] args) {          MahasiswaBerprestasi03 list = new MahasiswaBerprestasi03();          Mahasiswa03 m1 = new Mahasiswa03("123", "Zidan", "2A", 3.2);          Mahasiswa03 m2 = new Mahasiswa03("124", "Ayu", "2A", 3.5);          Mahasiswa03 m3 = new Mahasiswa03("125", "Sofi", "2A", 3.1);          Mahasiswa03 m4 = new Mahasiswa03("126", "Sita", "2A", 3.9);          Mahasiswa03 m5 = new Mahasiswa03("127", "Miki", "2A", 3.7);          list.tambah(m1);          list.tambah(m2);          list.tambah(m3);          list.tambah(m4);          list.tambah(m5);            System.out.println("data mahasiswa sebelum sorting : ");          list.tampil();          System.out.println("Data mahasiswa setelah sorting berdasarkan IPK (DESC) : ");          list.bubbleSort();          list.tampil();      }  } |

Hasil kode



**Pertanyaan**

1. **Jawaban**
   1. Karena dalam algoritma Bubble Sort, proses perbandingan antar elemen dilakukan sebanyak n - 1 kali (dimana n adalah jumlah elemen). Setelah n-1 tahap, data sudah pasti terurut sehingga tidak perlu melakukan perulangan ke-n.
   2. Agar proses perbandingan tidak perlu dilakukan terhadap elemen yang sudah berada di posisi akhir dan sudah pasti benar (terurut). Setiap iterasi luar (i) akan menempatkan elemen terbesar ke posisi akhir, sehingga perbandingan bisa dikurangi satu elemen di setiap iterasi.
   3. Karena i < 50 - 1, maka perulangan i akan terjadi sebanyak 49 kali.
2. **Modifikasi**

**Kode program**

**Kode program Mahasiswa03**

|  |
| --- |
| public class Mahasiswa03 {        String nim;      String nama;      String kelas;      double ipk;      public Mahasiswa03() {        }      public Mahasiswa03(String nm, String name, String kls, double ip) {          nim = nm;          nama = name;          kelas = kls;          ipk = ip;      }      void tampilInformasi(){          System.out.println("Nama : " + nama);          System.out.println("NIM : " + nim);          System.out.println("Kelas : " + kelas);          System.out.println("ipk : " + ipk);      }  } |

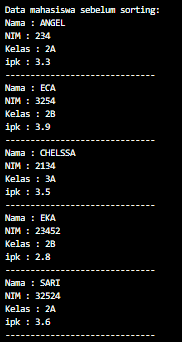
**Kode program MahasiswaBerprestasi03**

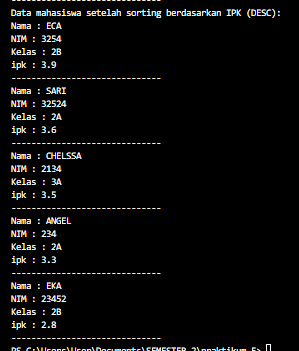
|  |
| --- |
| public class MahasiswaBerprestasi03 {      Mahasiswa03[] listMhs = new Mahasiswa03[5];      int idx;        void tambah (Mahasiswa03 m){          if (idx < listMhs.length){              listMhs[idx] = m;              idx++;          }else{              System.out.println("Data Sudah Penuh");          }      }        void tampil(){          for (Mahasiswa03 m:listMhs){              m.tampilInformasi();              System.out.println("------------------------------");          }      }        void bubbleSort(){          for (int i = 0; i < listMhs.length-1; i++){              for ( int j = 1; j < listMhs.length-i; j++){                  if (listMhs[j].ipk > listMhs[j-1].ipk){                      Mahasiswa03 tmp = listMhs[j];                      listMhs[j] = listMhs[j-1];                      listMhs[j-1] = tmp;                  }              }          }      }  } |

**Kode program MahasiswaDemo03**

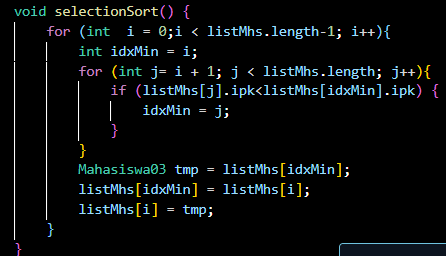
|  |
| --- |
| import java.util.Scanner;    public class MahasiswaDemo03 {      public static void main(String[] args) {          Scanner sc = new Scanner(System.in);          MahasiswaBerprestasi03 list = new MahasiswaBerprestasi03();            for (int i = 0; i < 5; i++) {              System.out.println("Masukkan Mahasiswa ke-" + (i+1) +":");              System.out.print("NIM: ");              String nim = sc.nextLine();              System.out.print("Nama: ");              String nama = sc.nextLine();              System.out.print("Kelas: ");              String kelas = sc.nextLine();              System.out.print("IPK: ");              double ipk = sc.nextDouble();              System.out.println("--------------------------------");              sc.nextLine();                Mahasiswa03 m = new Mahasiswa03(nim, nama, kelas, ipk);              list.tambah(m);            }          System.out.println("\nData mahasiswa sebelum sorting: ");          list.tampil();            System.out.println("Data mahasiswa setelah sorting berdasarkan IPK (DESC): ");          list.bubbleSort();          list.tampil();            sc.close();      }    } |

**Hasil kode**

** **

****

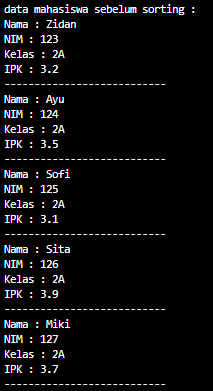
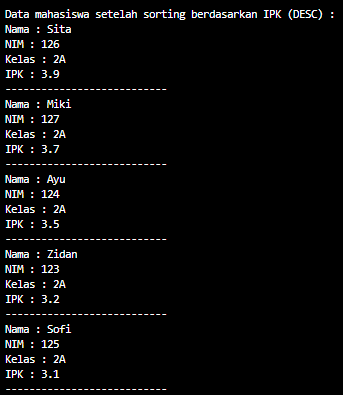
1. **Mengurutkan data**
2. Lihat kembali class MahasiswaBerprestasi, dan tambahkan method selectionSort() di dalamnya! Method ini juga akan melakukan proses sorting secara ascending, tetapi menggunakan pendekatan selection sort.

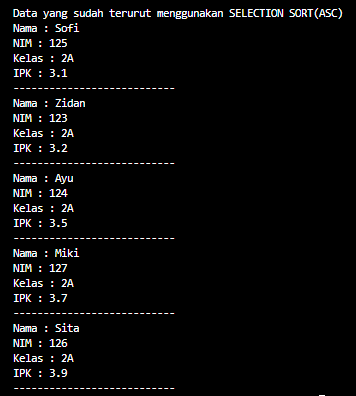


1. Setelah itu, buka kembali class MahasiswaDemo, dan di dalam method main() tambahkan baris program untuk memanggil method selectionSort() tersebut, kemudian panggil method tampil() untuk menampilkan data yang sudah diurutkan!

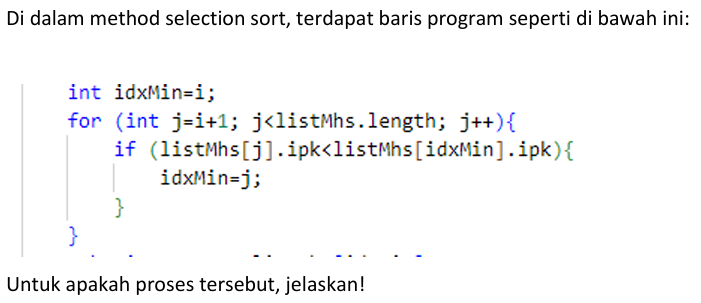


**Hasil kode**



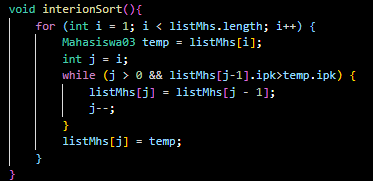
**Pertanyaan**



**Jawaban :**

Menentukan posisi data dengan nilai IPK terkecil dari elemen ke-I hingga akhir. idxMin akan menyimpan index elemen dengan IPK terkecil selama proses pencarian berlangsung. Maka kode tersebut digunakan untuk mencari index elemen dengan IPK terkecil, Sebagian dari algoritma selection sort

1. **Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort**
2. Lihat kembali class MahasiswaBerprestasi, dan tambahkan method insertionSort() didalamnya. Method ini juga akan melakukan proses sorting secara ascending, tetapi menggunakan pendekatan Insertion Sort.



1. Setelah itu, buka kembali class MahasiswaDemo, dan di dalam method main() tambahkan baris program untuk memanggil method insertionSort() dan tampil () tersebut!

**kode program**

**kode program Mahasiswa03.java**

|  |
| --- |
| public class Mahasiswa03 {      String nim;      String nama;      String kelas;      double ipk;      Mahasiswa03() {      }      Mahasiswa03(String nm, String nma, String kls, double ip) {          nim = nm;          nama = nma;          kelas = kls;          ipk = ip;      }      void tampilInformasi() {          System.out.println("Nama : " + nama);          System.out.println("NIM : " + nim);          System.out.println("Kelas : " + kelas);          System.out.println("IPK : " + ipk);      }  } |

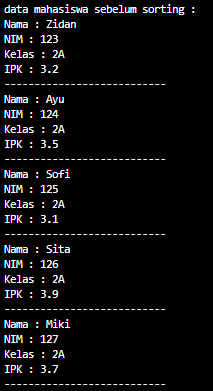
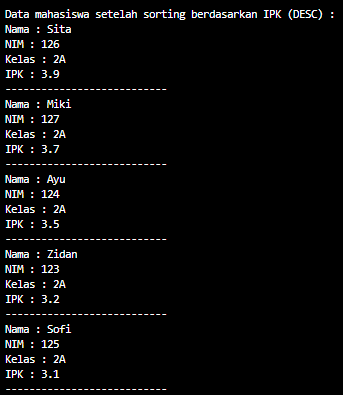
**kode program MahasiswaBerprestasi03.java**

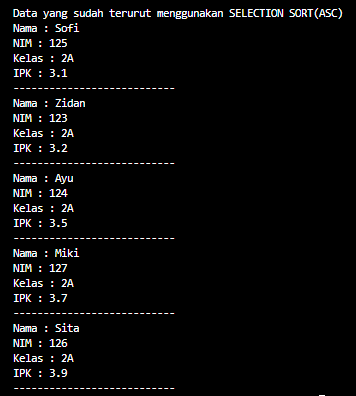
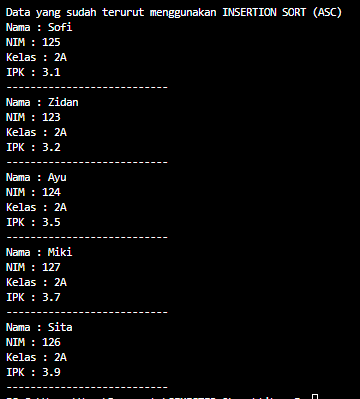
|  |
| --- |
| public class MahasiswaBerprestasi03 {      Mahasiswa03 [] listMhs = new Mahasiswa03[5];      int idx;      void tambah (Mahasiswa03 m) {          if (idx<listMhs.length) {              listMhs[idx] = m;              idx++;          }else {              System.out.println("data sudah penuh");          }      }      void tampil() {          for (Mahasiswa03 m:listMhs) {              m.tampilInformasi();              System.out.println("---------------------------");          }      }      void bubbleSort() {          for (int i = 0; i < listMhs.length-1; i++) {              for (int j = 1; j < listMhs.length-i; j++) {                  if (listMhs[j].ipk>listMhs[j-1].ipk) {                      Mahasiswa03 tmp =listMhs[j];                      listMhs[j] = listMhs[j-1];                      listMhs[j-1]  = tmp;                  }              }          }      }      void selectionSort() {          for (int  i = 0;i < listMhs.length-1; i++){              int idxMin = i;              for (int j= i + 1; j < listMhs.length; j++){                  if (listMhs[j].ipk<listMhs[idxMin].ipk) {                      idxMin = j;                  }              }              Mahasiswa03 tmp = listMhs[idxMin];              listMhs[idxMin] = listMhs[i];              listMhs[i] = tmp;          }      }      void interionSort(){          for (int i = 1; i < listMhs.length; i++) {              Mahasiswa03 temp = listMhs[i];              int j = i;              while (j > 0 && listMhs[j-1].ipk < temp.ipk) {                  listMhs[j] = listMhs[j - 1];                  j--;              }              listMhs[j] = temp;          }      }  } |

**kode program MahasiswaDemo03**

|  |
| --- |
| public class MahasiswaDemo03 {      public static void main(String[] args) {          MahasiswaBerprestasi03 list = new MahasiswaBerprestasi03();          Mahasiswa03 m1 = new Mahasiswa03("123", "Zidan", "2A", 3.2);          Mahasiswa03 m2 = new Mahasiswa03("124", "Ayu", "2A", 3.5);          Mahasiswa03 m3 = new Mahasiswa03("125", "Sofi", "2A", 3.1);          Mahasiswa03 m4 = new Mahasiswa03("126", "Sita", "2A", 3.9);          Mahasiswa03 m5 = new Mahasiswa03("127", "Miki", "2A", 3.7);          list.tambah(m1);          list.tambah(m2);          list.tambah(m3);          list.tambah(m4);          list.tambah(m5);            System.out.println("data mahasiswa sebelum sorting : ");          list.tampil();          System.out.println("Data mahasiswa setelah sorting berdasarkan IPK (DESC) : ");          list.bubbleSort();          list.tampil();          System.out.println("Data yang sudah terurut menggunakan SELECTION SORT(ASC)");          list.selectionSort();          list.tampil();          System.out.println("Data yang sudah terurut menggunakan INSERTION SORT (ASC)");          list.interionSort();          list.tampil();      }  } |

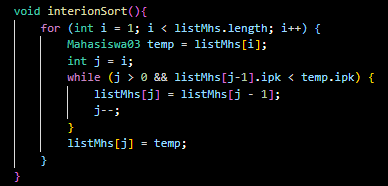
**Hasil**

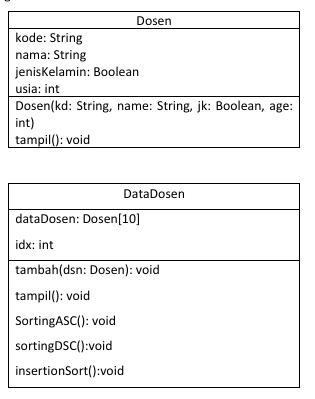
 

**Pertanyaan**

Ubahlah fungsi pada InsertionSort sehingga fungsi ini dapat melaksanakan proses sorting dengan cara descending.



**Latihan praktikum**



Berdasarkan class diagram diatas buatlah menu dikelas main dengan pilihan menu:

1. Tambah data digunakan untuk menambahkan data dosen
2. Tampil data digunakan untuk menampilkan data seluruh dosen
3. Sorting ASC digunakan untuk mengurutkan data dosen berdasarkan usia dimulai dari dosen termuda ke dosen tertua menggunakan bublle Sort.
4. Sorting DSC digunakan untuk mengurutkan data dosen berdasarkan usia dimulai dari tertua ke dosen termuda dapat menggunakan algoritma selection sort atau insertion sort.

**Kode program**

**Kode program Dosen.java**

|  |
| --- |
| public class Dosen03 {      String kode;      String nama;      boolean jenisKelamin;      int usia;      Dosen03(String kd, String name, boolean jk, int age) {          kode = kd;          nama = name;          jenisKelamin = jk;          usia = age;      }      void tampil() {          System.out.println("Kode           : " + kode);          System.out.println("Nama           : " + nama);          System.out.println("Jenis Kelamin  : " + (jenisKelamin ? "Laki-laki" : "Perempuan"));          System.out.println("Usia           : " + usia);          System.out.println("----------------------------------");      }  } |

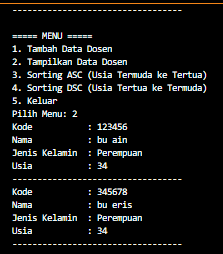
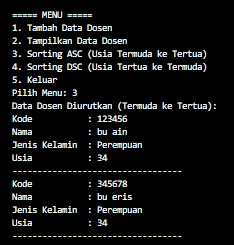
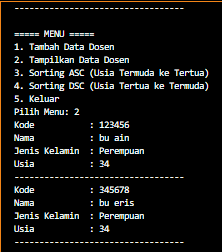
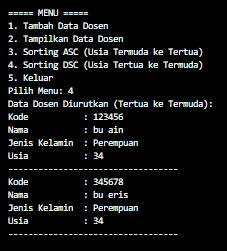
**Kode program DataDosen03.java**

|  |
| --- |
| public class DataDosen03 {      Dosen03[] dataDosen = new Dosen03[10];      int idx;      void tambah(Dosen03 dsn) {          if (idx < dataDosen.length) {              dataDosen[idx++] = dsn;          } else {              System.out.println("Data Dosen Sudah Penuh!");          }      }      void tampil() {          if (idx == 0) {              System.out.println("Tidak Ada Data Dosen.");              return;          }          for (int i = 0; i < idx; i++) {              dataDosen[i].tampil();          }      }      void sortingASC() {          for (int i = 0; i < idx - 1; i++) {              for (int j = 0; j < idx - 1 - i; j++) {                  if (dataDosen[j].usia > dataDosen[j + 1].usia) {                      Dosen03 temp = dataDosen[j];                      dataDosen[j] = dataDosen[j + 1];                      dataDosen[j + 1] = temp;                  }              }          }      }      void sortingDSC() {          for (int i = 0; i < idx - 1; i++) {              for (int j = 0; j < idx - 1 - i; j++) {                  if (dataDosen[j].usia < dataDosen[j + 1].usia) {                      Dosen03 temp = dataDosen[j];                      dataDosen[j] = dataDosen[j + 1];                      dataDosen[j + 1] = temp;                  }              }          }      }  } |

**Kode program DosenMain03.java**

|  |
| --- |
| import java.util.Scanner;  public class DosenMain03 {      public static void main(String[] args) {          Scanner sc = new Scanner(System.in);          DataDosen03 datadsn = new DataDosen03();          int pilihan;          do {              System.out.println("\n===== MENU =====");              System.out.println("1. Tambah Data Dosen");              System.out.println("2. Tampilkan Data Dosen");              System.out.println("3. Sorting ASC (Usia Termuda ke Tertua)");              System.out.println("4. Sorting DSC (Usia Tertua ke Termuda)");              System.out.println("5. Keluar");              System.out.print("Pilih Menu: ");              pilihan = sc.nextInt();              sc.nextLine();              switch (pilihan) {                  case 1:                      if (datadsn.idx >= 10) {                          System.out.println("Kapasitas data sudah penuh.");                          break;                      }                      System.out.print("Masukkan jumlah data yang ingin ditambahkan: ");                      int jumlah = sc.nextInt();                      sc.nextLine();                      for (int i = 0; i < jumlah; i++) {                          if (datadsn.idx >= 10) {                              System.out.println("Kapasitas data sudah penuh.");                              break;                          }                          System.out.println("Data Dosen ke-" + (datadsn.idx + 1));                          System.out.print("Kode Dosen       : ");                          String kode = sc.nextLine();                          System.out.print("Nama Dosen       : ");                          String nama = sc.nextLine();                          System.out.print("Jenis Kelamin (L/P): ");                          char jk = sc.next().toUpperCase().charAt(0);                          boolean jenisKelamin = jk == 'L';                          System.out.print("Usia             : ");                          int usia = sc.nextInt();                          sc.nextLine();                          Dosen03 dsn = new Dosen03(kode, nama, jenisKelamin, usia);                          datadsn.tambah(dsn);                          System.out.println("----------------------------------");                      }                      break;                  case 2:                      datadsn.tampil();                      break;                  case 3:                      datadsn.sortingASC();                      System.out.println("Data Dosen Diurutkan (Termuda ke Tertua):");                      datadsn.tampil();                      break;                  case 4:                      datadsn.sortingDSC();                      System.out.println("Data Dosen Diurutkan (Tertua ke Termuda):");                      datadsn.tampil();                      break;                  case 5:                      System.out.println("Keluar dari program.");                      break;                  default:                      System.out.println("Pilihan tidak valid. Coba lagi.");              }          } while (pilihan != 5);          sc.close();      }  } |

**Hasil kode**

**  **